IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 09/847,535
Filing Date 5/01/2001
First-Named Inventor ..... Bond
Applicant..... Microsoft Corporation
Attorney's Docket No. MS1-0665us
Title: *Kernel Emulator for Non-Native Program Modules*

Examiner	Phone	Fax	Office	Office Description
STEVENS THOMAS H	(571)272-3715		P/2123	GROUP ART UNIT 2123

APPEAL BRIEF

To: MS: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

From: Kasey C. Christie (Tel. 509-324-9256; Fax 509-323-8979)
Customer No. 22801

Pursuant to 37 C.F.R. §1.192, Applicant hereby submits an appeal brief for Application No. 09/847,535. A Notice of Appeal was filed December 12, 2005. Accordingly, Applicant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

421 West Riverside, Suite 500
Spokane, WA 99201
T: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee&hayes

TABLE OF CONTENTS

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee&hayes

(1) Real Party in Interest

The real party in interest is the Microsoft Corporation, the assignee of all right and title to the subject invention.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

(2) Related Appeals, Interferences, and Judicial Proceedings

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee&hayes

(3) Status of Claims

Claims 1-42 and 45-46 are pending in this Application, and are set forth in the Appendix of Appealed Claims on page 31. All pending claims (claims 1-42 and 45-46) stand rejected. Claims 1-46 were originally filed in the Application. No claims have been allowed. In response to a restriction requirement, claims 43 and 44 were non-elected and thus withdrawn from consideration without prejudice. Claims 10, 29, and 42 have been amended.

Claims 1-42 and 45-46 are subject of this appeal and stand rejected as set forth in a Final Office Action dated July 13, 2005 (hereinafter, the "FINAL ACTION").

Specifically, the Office rejects:

- Claims 1, 3-6, 9-13, 15-17, 19-28, 34, 37-40, 42, and 45-46 under USC § 102(b) as being anticipated by **Scalzi** (*Scalzi et al.*, US Patent No. 5,560,013 (issued 9/24/1996)), as set forth in pp. 5-7 of the FINAL ACTION;
- Claims 2 and 14 under USC § 103(a) as being unpatentable over **Scalzi** in view of **Franz** (Michael Franz, "Emulating an Operating System on Top of Another" Software – Practice and Experience. Vol. 23, No. 6, June 1993, pp. 677-692), as set forth in p. 8 of the FINAL ACTION;
- Claims 7, 8, 18, 35, and 41 under USC § 103(a) as being unpatentable over **Scalzi** in view of **Duvall** (*Duvall et al.*, US Patent No. 4,742,447 (issued 5/3/1988)), as set forth in pp. 9-10 of the FINAL ACTION;

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

- Claim 36 under USC § 103(a) as being unpatentable over **Scalzi** in view of **Duvall** and further in view of **Franz,** as set forth in pp. 10-11 of the FINAL ACTION; and
- Claims 29-33 under USC § 103(a) as being unpatentable over **Duvall** in view of **McCoy** (*McCoy et al.,* US Patent No. 5,036,484 (issued 7/30/1991)), as set forth in pp. 11-13 of the FINAL ACTION.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee hayes

(4) Status of Amendments

The Applicant responded to a non-final Office Action issued on November 30, 2004 (hereinafter, the "NON-FINAL ACTION"). In that response, Applicant elected claims 1-42 and 45-46 and amended claim 10, 29, and 42. Applicant traversed all substantive rejections.

After that, the FINAL ACTION issued on July 13, 2005—the action dismissing Applicant's traversal and maintaining the rejection of all pending claims. In Applicant's response to the FINAL ACTION, Applicant traversed all substantive rejections and amended no claims. No other amendments have been filed subsequent to the FINAL ACTION.

The Office issued an advisory action dated December 8, 2005 (hereinafter, the "ADVISORY ACTION") that dismissed Applicant's traversal and maintained the rejection of all pending claims. No other amendments have been filed subsequent to the FINAL ACTION or ADVISORY ACTION.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

(5) Summary of Claimed Subject Matter

Broadly speaking, the claimed subject matter describes a technology facilitating the operation of non-native program modules within a native computing platform. More particularly, the described technology facilitates the interoperability of native and non-native program modules within a native computing platform.

This technology involves an emulation of the kernel of the non-native operating system. Instead of interacting with the native kernel of the native computing platform, the non-native program modules interact with a non-native kernel emulator. This emulator handles the necessary conversions and translations. With this non-native kernel emulation, native and non-native program modules are interoperable. Except for the kernel emulator, none of the program module (native or non-native) and none of the other portions of the native computing platform are aware of the emulation. The computing environment and other program modules appear to be non-native to the non-native program modules. Likewise, the non-native program modules appear to be native to the computing environment and the native program modules.

Following is a concise explanation of each independent claim 1, 13, 29, 34, 40, and 45 involved in the Appeal which includes specification references and exemplary drawing reference characters. As explained, the independent claims are not limited solely to the elements identified by the reference characters.

Specifically:

Claim 1 includes an interceptor (400) configured to intercept kernel calls from non-native program modules and a call-converter (412, 414, and 416)

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee & hayes

configured to convert non-native kernel calls intercepted by the interceptor into native kernel calls.

Claim 13 includes intercepting (512 and 514) kernel calls from non-native program modules and converting (516, 518, 519, 550, and 552) the intercepted non-native kernel calls into native kernel calls.

Claim 29 includes determining (512) whether an initiating program module is a native or non-native; if the initiating program is non-native; limiting (516) available memory to a range that is addressable by the non-native program module, that range of addressable memory being less that the available memory; establishing (518) non-native a version of a shared memory data structure that may be synchronized with a native version of the same shared memory data structure.

Claim 34 includes emulating (Figs. 5A and 5B) a non-native kernel for a native computing platform so that kernel calls from non-native applications are translated into calls to a native kernel.

Claim 40 includes a kernel emulator (400) configured to emulate a non-native kernel for a native computing platform so that kernel calls from non-native applications are translated into calls to a native kernel.

Claim 45 includes an interceptor (400) configured to intercept kernel calls from non-native program modules; a call-converter (412, 414, and 416) configured to convert non-native kernel calls intercepted by the interceptor into native kernel calls, wherein the call-converter comprises: an instruction-translator (412) configured to translate non-native CPU instructions into native CPU instructions; an address-translator (414) configured to translate addresses from non-native length into native length.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee hayes

(6) Grounds of Rejection to be Reviewed on Appeal

A. Whether **Scalzi** anticipates claims 1, 3-6, 9-13, 15-17, 19-28, 34, 37-40, 42, and 45-46 under 35 U.S.C. § 102(b) and whether the Office has satisfactorily met its burden to show such anticipation?

B. Whether claims 2 and 14 are obvious under USC § 103(a) based upon the combination of **Scalzi** and **Franz** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

C. Whether claims 7, 8, 18, 35, and 41 are obvious under USC § 103(a) based upon the combination of **Scalzi** and **Duvall** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

D. Whether claim 36 is obvious under USC § 103(a) based upon the combination of **Scalzi, Duvall,** and **Franz** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

E. Whether claims 29-33 are obvious under USC § 103(a) based upon the combination of **Duvall** and **McCoy** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

lee&hayes
421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com

(7) Argument

Issue A -- Whether **Scalzi** anticipates claims 1, 3-6, 9-13, 15-17, 19-28, 34, 37-40, 42, and 45-46 under 35 U.S.C. § 102(b) and whether the Office has satisfactorily met its burden to show such anticipation?

Scalzi

Scalzi describes hardware emulation. In particular, it describes a method of utilizing large virtual addressing in a target computer to implement an instruction set translator (IST) for dynamically translating the machine language instructions of an alien source computer into a set of functionally equivalent target computer machine language instructions, providing in the target machine, an execution environment for source machine operating systems, application subsystems, and applications.

The target system provides a unique pointer table in target virtual address space that connects each source program instruction in the multiple source virtual address spaces to a target instruction translation which emulates the function of that source instruction in the target system. The target system stores the translated executable source programs by actually storing only one copy of any source program, regardless of the number of source address spaces in which the source program exists.

The target system manages dynamic changes in the source machine storage, accommodating the nature of a preemptive, multitasking source operating system. The target system preserves the security and data integrity for the source programs

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509 323-8979
www.leehayes.com
lee&hayes

on a par with their security and data integrity obtainable when executing in source processors (i.e. having the source architecture as their native architecture). The target computer execution maintains source-architected logical separations between programs and data executing in different source address spaces--without a need for the target system to be aware of the source virtual address spaces.

Claim 1

With portions of **Scalzi** which were cited by the Office in the FINAL ACTION provided in brackets, this claim recites (in part):

- an interceptor configured to intercept kernel calls from non-native program modules; **[Fig. 1, element 102 (the "emulator control program") and description]**
- a call-converter configured to convert non-native kernel calls intercepted by the interceptor into native kernel calls. **[Fig. 1, element 103 (the "instruction set translator") and description]**

In general, the subject-matter of this claim is "kernel emulation" and operating on or in response to kernel calls (e.g., application programming interfaces (APIs)). In short, Applicant respectfully submits that **Scalzi** does not anticipate this claim because **Scalzi** discloses conventional "hardware emulation" instead of "kernel emulation" and acting on or in response to kernel calls.

Furthermore, **Scalzi** never discloses emulating a kernel of an operating system. Further still, **Scalzi** never even mentions a kernel of an operating system. Of course, Applicant understands that it is possible for **Scalzi** to address these

lee&hayes
421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com

concepts without actually mentioning them by name. However, Applicant submits that **Scalzi** does not address the concepts related to "kernel emulation" because it is focused on emulation of the actual hardware and not kernel emulation.

It is easy to confuse "hardware" and "kernel" emulation. That is because the kernel (and the OS itself) is inextricably linked to the hardware architecture. However, the two are not the same. While kernel emulation is depending upon the underlying hardware architecture, hardware emulation does not necessitate an emulation of the kernel.

The architectural configuration and characteristics of conventional hardware emulation approaches (e.g., Virtual Machine or VM) are shown in Fig. 2 and discussed on pages 7-8 of the Application. The architectural configuration of kernel emulation is shown in Fig. 3 and discussed on pages 13-15 of the Application.

Moreover, Applicant submits that **Scalzi** does not disclose the interception of a kernel call. However, this claim recites an interception of a kernel call. Since **Scalzi** never mentions kernel calls, and thus, never discloses intercepting such kernel calls. Since **Scalzi** emulates hardware, it does not need to handle kernel calls.

While it appears that **Scalzi** does disclose instructions conversion, Applicant submits that **Scalzi** fails to disclose kernel emulation, kernel calls, or interception of such kernel calls. As shown above, **Scalzi** does not disclose all of the claimed elements and features of this claim.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee&hayes

Claims 2-12

These claims ultimately depend upon independent claim 1. As discussed above, claim 1 is allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

Claim 13

With portions of **Scalzi** which were cited by the Office in the FINAL ACTION provided in brackets, this claim recites (in part):

- intercepting kernel calls from non-native program modules; **[Fig. 1, element 102 (the "emulator control program") and description]**
- converting the intercepted non-native kernel calls into native kernel calls. **[Fig. 1, element 103 (the "instruction set translator") and description]**

In general, the subject-matter of this claim is "kernel emulation" and operating on or in response to kernel calls (e.g., application programming interfaces (APIs)). In short, Applicant respectfully submits that **Scalzi** does not anticipate this claim because **Scalzi** discloses conventional "hardware emulation" instead of "kernel emulation" and acting on or in response to kernel calls.

Furthermore, **Scalzi** never discloses emulating a kernel of an operating system. Further still, **Scalzi** never even mentions a kernel of an operating system. Of course, Applicant understands that it is possible for **Scalzi** to address these concepts without actually mentioning them by name. However, Applicant submits that **Scalzi** does not address the concepts related to "kernel emulation" because it is focused on emulation of the actual hardware and not kernel emulation.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee&hayes

It is easy to confuse "hardware" and "kernel" emulation. That is because the kernel (and the OS itself) is inextricably linked to the hardware architecture. The architectural configuration and characteristics of conventional hardware emulation approaches (e.g., Virtual Machine or VM) are shown in Fig. 2 and discussed on pages 7-8 of the Application. The architectural configuration of kernel emulation is shown in Fig. 3 and discussed on pages 13-15 of the Application.

Moreover, Applicant submits that **Scalzi** does not disclose the interception of a kernel call. However, this claim recites an interception of a kernel call. Since **Scalzi** never mentions kernel calls, and thus, never discloses intercepting such kernel calls.

While it appears that **Scalzi** does disclose instructions conversion, Applicant submits that **Scalzi** fails to disclose kernel emulation, kernel calls, or interception of such kernel calls. As shown above, **Scalzi** does not disclose all of the claimed elements and features of this claim.

Claims 14-28

These claims ultimately depend upon independent claim 13. As discussed above, claim 13 is allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee&hayes

Claim 34

With portions of **Scalzi** which were cited by the Office in the FINAL ACTION provided in brackets, this claim recites (in part):

- emulating a non-native kernel for a native computing platform so that kernel calls from non-native applications are translated into calls to a native kernel. **[Fig. 1, element 103 (the "instruction set translator") and description]**

In general, the subject-matter of this claim is "kernel emulation" and operating on or in response to kernel calls (e.g., application programming interfaces (APIs)). In short, Applicant respectfully submits that **Scalzi** does not anticipate this claim because **Scalzi** discloses conventional "hardware emulation" instead of "kernel emulation" and acting on or in response to kernel calls.

Furthermore, **Scalzi** never discloses emulating a kernel of an operating system. Further still, **Scalzi** never even mentions a kernel of an operating system. Of course, Applicant understands that it is possible for **Scalzi** to address these concepts without actually mentioning them by name. However, Applicant submits that **Scalzi** does not address the concepts related to "kernel emulation" because it is focused on emulating the actual hardware.

It is easy to confuse "hardware" and "kernel" emulation. That is because the kernel (and the OS itself) is inextricably linked to the hardware architecture. The architectural configuration and characteristics of conventional hardware emulation approaches (e.g., Virtual Machine or VM) are shown in Fig. 2 and discussed on pages 7-8 of the Application. The architectural configuration of

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee&hayes

kernel emulation is shown in Fig. 3 and discussed on pages 13-15 of the Application.

Moreover, Applicant submits that **Scalzi** does not disclose translating "kernel calls from non-native applications" into "calls to a native kernel." However, this claim recites a kernel emulation that translates "kernel calls from non-native applications" into "calls to a native kernel." Since **Scalzi** never mentions kernel calls, and thus, never discloses translating such kernel calls.

While it appears that **Scalzi** does disclose instructions conversion, Applicant submits that **Scalzi** fails to disclose kernel emulation or translation of such kernel calls. As shown above, **Scalzi** does not disclose all of the claimed elements and features of this claim.

Claims 35-39

These claims ultimately depend upon independent claim 34. As discussed above, claim 34 is allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

Claim 40

With portions of **Scalzi** which were cited by the Office in the FINAL ACTION provided in brackets, this claim recites (in part):

- a kernel emulator configured to emulate a non-native kernel for a native computing platform so that kernel calls from non-native applications are translated into calls to a native kernel. **[Fig. 1, element 103 (the "instruction set translator") and description]**

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee&hayes

In general, the subject-matter of this claim is "kernel emulation" and operating on or in response to kernel calls (e.g., application programming interfaces (APIs)). In short, Applicant respectfully submits that **Scalzi** does not anticipate this claim because **Scalzi** discloses conventional "hardware emulation" instead of "kernel emulation" and acting on or in response to kernel calls.

Furthermore, **Scalzi** never discloses emulating a kernel of an operating system. Further still, **Scalzi** never even mentions a kernel of an operating system. Of course, Applicant understands that it is possible for **Scalzi** to address these concepts without actually mentioning them by name. However, Applicant submits that **Scalzi** does not address the concepts related to "kernel emulation" because it is focused on emulating the actual hardware.

It is easy to confuse "hardware" and "kernel" emulation. That is because the kernel (and the OS itself) is inextricably linked to the hardware architecture. The architectural configuration and characteristics of conventional hardware emulation approaches (e.g., Virtual Machine or VM) are shown in Fig. 2 and discussed on pages 7-8 of the Application. The architectural configuration of kernel emulation is shown in Fig. 3 and discussed on pages 13-15 of the Application.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee&hayes

Moreover, Applicant submits that **Scalzi** does not disclose translating "kernel calls from non-native applications" into "calls to a native kernel." However, this claim recites a kernel emulation that translates "kernel calls from non-native applications" into "calls to a native kernel." Since **Scalzi** never mentions kernel calls, and thus, never discloses translating such kernel calls.

While it appears that **Scalzi** does disclose instructions conversion, Applicant submits that **Scalzi** fails to disclose kernel emulation or translation of such kernel calls. As shown above, **Scalzi** does not disclose all of the claimed elements and features of this claim.

Claims 41 and 42

These claims ultimately depend upon independent claim 40. As discussed above, claim 40 is allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

Claim 45

With portions of **Scalzi** which were cited by the Office in the FINAL ACTION provided in brackets, this claim recites (in part):

- an interceptor configured to intercept kernel calls from non-native program modules; **[Fig. 1, element 102 (the "emulator control program") and description]**
- a call-converter configured to convert non-native kernel calls intercepted by the interceptor into native kernel calls **[Fig. 1,**

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
leehayes

element 103 (the "instruction set translator") and description], wherein the call-converter comprises:

- an instruction-translator configured to translate non-native CPU instructions into native CPU instructions; **[Fig. 1, element 103 (the "instruction set translator") and description]**
- an address-translator configured to translate addresses from non-native length into native length. **[Fig. 3 and description, col. 21, lines 42-48]**

In general, the subject-matter of this claim is "kernel emulation" and operating on or in response to kernel calls (e.g., application programming interfaces (APIs)). In short, Applicant respectfully submits that **Scalzi** does not anticipate this claim because **Scalzi** discloses conventional "hardware emulation" instead of "kernel emulation" and acting on or in response to kernel calls.

Furthermore, **Scalzi** never discloses emulating a kernel of an operating system. Further still, **Scalzi** never even mentions a kernel of an operating system. Of course, Applicant understands that it is possible for **Scalzi** to address these concepts without actually mentioning them by name. However, Applicant submits that **Scalzi** does not address the concepts related to "kernel emulation" because it is focused on emulation of the actual hardware and not kernel emulation.

It is easy to confuse "hardware" and "kernel" emulation. That is because the kernel (and the OS itself) is inextricably linked to the hardware architecture. The architectural configuration and characteristics of conventional hardware emulation approaches (e.g., Virtual Machine or VM) are shown in Fig. 2 and

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

discussed on pages 7-8 of the Application. The architectural configuration of kernel emulation is shown in Fig. 3 and discussed on pages 13-15 of the Application.

Moreover, Applicant submits that **Scalzi** does not disclose the interception of a kernel call. However, this claim recites an interception of a kernel call. Since **Scalzi** never mentions kernel calls, and thus, never discloses intercepting such kernel calls.

While it appears that **Scalzi** does disclose instructions conversion, Applicant submits that **Scalzi** fails to disclose kernel emulation, kernel calls, or interception of such kernel calls. As shown above, **Scalzi** does not disclose all of the claimed elements and features of this claim.

Claim 46

This claim ultimately depends upon independent claim 45. As discussed above, claim 45 is allowable. In addition to its own merits, this dependent claim is allowable for the same reasons that its base claim is allowable.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee & hayes

Issue B -- Whether claims 2 and 14 are obvious under USC § 103(a) based upon the combination of **Scalzi** and **Franz** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

Claims 2 and 14

Claim 2 ultimately depends upon independent claim 1. As discussed above, claim 1 is allowable. Claim 14 ultimately depends upon independent claim 13. As discussed above, claim 13 is allowable.

In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee&hayes

Issue C -- Whether claims 7, 8, 18, 35, and 41 are obvious under USC § 103(a) based upon the combination of **Scalzi** and **Duvall** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

Claims 7, 8, 18, 35, and 41

Claims 7 and 8 ultimately depend upon independent claim 1. As discussed above, claim 1 is allowable. Claim 18 ultimately depends upon independent claim 13. As discussed above, claim 13 is allowable. Claim 35 ultimately depends upon independent claim 34. As discussed above, claim 34 is allowable. Claim 41 ultimately depends upon independent claim 40. As discussed above, claim 40 is allowable.

In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

Furthermore, Applicant submits the neither **Scalzi** nor **Duvall** discloses "a shared-memory manager configured to manage memory space that is accessible to both native and non-native program modules" like as is recited in claim 8.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
leeᵃhayes

Issue D – Whether claim 36 is obvious under USC § 103(a) based upon the combination of **Scalzi, Duvall,** and **Franz** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

Claim 36

Claim 36 ultimately depends upon independent claim 34. As discussed above, claim 34 is allowable. In addition to its own merits, this dependent claim is allowable for the same reasons that its base claim is allowable.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

Issue E -- Whether claims 29-33 are obvious under USC § 103(a) based upon the combination of **Duvall** and **McCoy** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

Claim 29

In the FINAL ACTION, the Office provides the following reasoning for rejecting this claim:

> 29. **Claims 29-33 are** rejected under 35 U.S.C. 103(a) as being unpatentable over **Duvall** and further in view of McCoy et al (U.S. Patent Number 5,036,484), herein referred to as **McCoy.**
>
> 30. **As to Claim 29, Duvall teaches:** a method comprising: if the initiating program is non-native: limiting available memory to a range that is addressable by the non-native program module **(column 4, lines 43-46, column 6, lines 25-29, column 9, lines 20-25)**; establishing non-native a version of a shared memory data structure that may be synchronized with a native version of the same shared memory data structure **(column 5, lines 45-51, column 6, lines 25-29)**.
>
> 31. Duvall further teaches the data in a segment of virtual memory is created as a result of an application program being run **(column 5, lines 52-55)**. While this implies that must be some determination as to weather a program is native or non-native allowing for the segment in virtual memory to be created, **Duvall** does not expressly teach determining whether an initiating program module is a native or non-native.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee@hayes

32. **McCoy** teaches determining whether an initiating program module is a native or non-native **(Figure 3a, element 36a, column 5, lines 40-48)** in a system that emulates a host program in a PC environment and translates host data to PC format by the emulation program **(column 5, lines 28-31)**, allowing the system to know whether to perform a function of the native system or perform a function of the non-native system which includes the translation of code **(Figure 3a, element 31a and column 5, lines 40-48)** since in the emulation systems of the prior art, when operating in emulation mode, the native system is incapable of performing functions other than those of the terminal which is being emulated. Therefore, the functions of the personal computer are not available in the emulation mode **(column 1, lines 32-39)**.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify since it would be advantageous to use McCoy to modify Duvall since a proliferation of many software programs has led to a lack of uniformity in the way in which data is formatted...and the presence of functions or operations which are unique to each program (column 1, lines 45-50), despite the fact that processors are operating at ever increasing speeds and efficiencies...nevertheless there's a finite delay between the time the operator request a document for conversion processing (column 2, lines 45-50).

Duvall discloses virtual machine (VM) technology, which the Applicant discusses in its Background section on p. 7 and 8 of the Application. **Duvall** discloses a new addressing scheme for VMs to use to read/write from/to a "file" (rather than memory). Duvall does not disclose "limiting available memory to a range that is addressable." Rather, it discloses a re-definition and re-arrangement of the meaning of the bits in the existing and unmodified addressable range.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

In the response to the NON-FINAL ACTION, Applicant amended this claim in the following manner to make it clear that the "limiting" has the effect of reducing the range of available memory that a non-native program module may address:

> limiting available memory to a range that is addressable by the non-native program module, that range of addressable memory being less that the available memory

Applicant submits that this above-identified amended language (which as was added in the the response to the NON-FINAL ACTION) has not yet been examined. The Office makes not reference in either the FINAL ACTION or the ADVISORY ACTION to this amended language. Furthermore, the Office has not cited anything found in any reference which discloses this recited language (that being: "that range of addressable memory being less that the available memory").

Moreover, the Office has not identified where **Duvall** discloses "non-native" program modules. Indeed, since **Duvall** discloses a VM model, then all program modules operating under a particular VM are presumptively native to that VM. If not, then an emulator would be necessary, but **Duvall** does not disclose an emulator.

While **McCoy** does disclose a nominal "emulator," it is not an emulation related to program modules being considered native or non-native. Rather, **McCoy** discloses a terminal emulation—that is, emulation of the operation of a "dumb" terminal connected to a host computer (e.g., mainframe computer).

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee&hayes

The Office indicates the **McCoy** discloses an initiation of a program module based upon a determination of whether a program is native or non-native. It points to col. 5, lines 40-48, Fig. 3a, element 36a, which is reproduced here:

> Keystrokes on the keyboard/display 35a are examined by the keystroke interpretation portion 36a of the emulation program to determine whether a PC or a host function is required. Program block 36a is responsive to the selected mode. In the PC mode, the keystrokes are handled by block 37a as normal keyboard commands or data. In the emulation mode, the keystrokes representing the host keys are passed to the host processor via the host emulator 31a.

However, this particular cited portion (and **McCoy** as a whole) are focused on determining from whence input (e.g., keystrokes) is received and processing them accordingly. The first sentence of the passage above says, "Keystrokes…are examined…to determine whether a PC or a host function is required." Applicant respectfully submits that this is not equivalent to "determining whether an initiating program module is a native or non-native."

Indeed, Applicant submits that all of **McCoy's** program modules (include the **McCoy's** terminal emulation program itself) are presumptively native. If they were non-native, then they would not function on the PC absent an operating-system based emulation program. However, **McCoy** does not disclose such an emulation program.

For the reasons given above, Applicant submits the combination of **Duvall** and **McCoy** fail to disclose all of the elements and features of this claim.

421 West Riverside, Suite 500
Spokane, WA 99201
T. 509.324.9256
F. 509.323.8979
www.leehayes.com
lee&hayes

No Motivation to Combine References

Furthermore, Applicant asserts that there is no motivation to combine the teachings of **Duvall** and the teachings of **McCoy**.

As discussed above, **Duvall** describes an addressing scheme for accessing files in a VM environment. However, **McCoy** describes "dumb" terminal emulation on a PC.

Applicant submits that there is no suggestion, teaching, or reason given by one reference that would motivate one of ordinary skill in the art at the time of the invention (hereinafter, "OOSA") to combine it with the teachings of the other reference. More importantly, Applicant submits that the Office has not provided any objective evidence showing why OOSA would be motivated to combine the teachings of the two references.

Duvall says nothing that would motivate OOSA to look towards **McCoy** and combine their teachings. Likewise, **McCoy** says nothing that would motivate OOSA to look towards **Duvall** and combine their teachings.

Accordingly, Applicant submits that OOSA would not be motivated to combine the VM file-access I/O addressing scheme of **Duvall** with the "dumb" terminal emulation of **McCoy**.

Claims 30-33

These claims ultimately depend upon independent claim 29. As discussed above, claim 29 is allowable.

In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable. Applicant submits that the

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee@hayes

Office withdraw the rejection of each of these dependent claims because its base claim is allowable.

Conclusion

Based upon the foregoing reasons, Applicant submits the **Scalzi** does not anticipate claims 1, 3-6, 9-13, 15-17, 19-28, 34, 37-40, 42, and 45-46 under 35 U.S.C. § 102(b) and/or the Office has satisfactorily met its burden to show such anticipation. Furthermore, claims 2, 7, 8, 14, 18, 29, 33, 35, 36, and 41 are not obvious under USC § 103(a) based upon the combination recited above for each rejected claim and/or the Office has satisfactorily met its burden to show that these rejected claims are obvious and that the combinations of references are proper?

Applicant respectfully requests that the outstanding rejections be overturned and that the pending claims 1-42 and 45-46 be allowed to issue.

Respectfully Submitted,

Dated: 3·13·06 By: _____

Kasey C. Christie
Reg. No. 40559
(509) 324-9256 x232
kasey@leehayes.com
www.leehayes.com

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee hayes

(8) Appendix of Appealed Claims

1. (ORIGINAL) A kernel emulator for non-native program modules, the emulator comprising:

an interceptor configured to intercept kernel calls from non-native program modules;

a call-converter configured to convert non-native kernel calls intercepted by the interceptor into native kernel calls.

2. (ORIGINAL) An emulator as recited in claim 1, wherein the call-converter comprises a translator configured to translate a non-native paradigm for passing parameters into a native paradigm for passing parameters.

3. (ORIGINAL) An emulator as recited in claim 1, wherein the call-converter comprises a translator configured to translate non-native CPU instructions into native CPU instructions.

4. (ORIGINAL) An emulator as recited in claim 1, wherein the call-converter comprises a translator configured to translate addresses from non-native length into native length.

5. (ORIGINAL) An emulator as recited in claim 1, wherein the call-converter comprises an argument-converter configured to convert non-native argument format into native argument format.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee@hayes

6. (ORIGINAL) An emulator as recited in claim 1, wherein the call-converter comprises a translator configured to translate words from non-native word size into native word size.

7. (ORIGINAL) An emulator as recited in claim 1 further comprising a memory constrainer configured to limit addressable memory to a range addressable by non-native program modules.

8. (ORIGINAL) An emulator as recited in claim 1 further comprising a shared-memory manager configured to manage memory space that is accessible to both native and non-native program modules.

9. (ORIGINAL) An emulator as recited in claim 1 further comprising a shared-memory manager configured to synchronize a native shared data structure with a non-native shared data structure.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

10. (**PREVIOUSLY PENDING**) An emulator as recited in claim 1 further comprising a shared-memory manager configured to manage memory space that is accessible to both native and non-native program modules, wherein the shared-memory manager maps versions of process shared data structures (process SDSs) and versions of thread shared data structures (thread SDSs) between native and non-native program modules.

11. (**ORIGINAL**) An operating system on a computer-readable medium, comprising:

a native kernel configured to receive calls from native program modules;

a kernel emulator as recited in claim 1 configured to receive calls from non-native program modules.

12. (**ORIGINAL**) An operating system on a computer-readable medium, comprising:

a native kernel configured to receive calls from native APIs;

a kernel emulator as recited in claim 1 configured to receive calls from non-native APIs.

13. (**ORIGINAL**) A method of emulating a kernel for non-native program modules, the method comprising:

intercepting kernel calls from non-native program modules;

converting the intercepted non-native kernel calls into native kernel calls.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee&hayes

14. (ORIGINAL) A method as recited in claim 13, wherein the converting step comprises translating a non-native paradigm for passing parameters into a native paradigm for passing parameters.

15. (ORIGINAL) A method as recited in claim 13, wherein the converting step comprises translating non-native CPU instructions into native CPU instructions.

16. (ORIGINAL) A method as recited in claim 13, wherein the converting step comprises translating addresses from non-native length into native length.

17. (ORIGINAL) A method as recited in claim 13, wherein the converting step comprises translating words from non-native word size into native word size.

18. (ORIGINAL) A method as recited in claim 13 further comprising limiting addressable memory to a range addressable by non-native program modules.

19. (ORIGINAL) A method as recited in claim 13 further comprising synchronizing a native shared data structure with a non-native shared data structure.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee&hayes

20. (ORIGINAL) A method as recited in claim 13 further comprising mapping versions of process shared data structures (SDSs) between native and non-native program modules.

21. (ORIGINAL) A method as recited in claim 19, wherein a process SDS of a native program module includes a pointer to a process SDS of a non-native program module.

22. (ORIGINAL) A method as recited in claim 19, wherein a process SDS of a non-native program module includes a pointer to a process SDS of a native program module.

23. (ORIGINAL) A method as recited in claim 13 further comprising mapping versions of thread shared data structures (SDSs) data structure between native and non-native program modules.

24. (ORIGINAL) A method as recited in claim 22, wherein a thread SDS of a native program module includes a pointer to a thread SDS of a non-native program module.

25. (ORIGINAL) A method as recited in claim 22, wherein a thread SDS of a non-native program module includes a pointer to a thread SDS of a native program module.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com

26. (**ORIGINAL**) A computer comprising one or more computer-readable media having computer-executable instructions that, when executed by the computer, perform the method as recited in claim 13.

27. (**ORIGINAL**) A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 13.

28. (**ORIGINAL**) An operating system embodied on a computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 13.

29. (**PREVIOUSLY PENDING**) A method comprising:

determining whether an initiating program module is a native or non-native;

if the initiating program is non-native:

limiting available memory to a range that is addressable by the non-native program module, that range of addressable memory being less that the available memory;

establishing non-native a version of a shared memory data structure that may be synchronized with a native version of the same shared memory data structure.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee&hayes

30. (ORIGINAL) A method as recited in claim 29 further comprising:

intercepting kernel calls from the non-native program module;

converting the intercepted non-native kernel calls into native kernel calls.

31. (ORIGINAL) A method as recited in claim 29 further comprising emulating a non-native kernel for which kernel calls from the non-native program module are intended.

32. (ORIGINAL) A computer comprising one or more computer-readable media having computer-executable instructions that, when executed by the computer, perform the method as recited in claim 29.

33. (ORIGINAL) A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 29.

34. (ORIGINAL) A method comprising emulating a non-native kernel for a native computing platform so that kernel calls from non-native applications are translated into calls to a native kernel.

35. (ORIGINAL) A method as recited in claim 34, wherein the emulating step comprises:

translating non-native CPU instructions into native CPU instructions;

translating addresses from non-native length into native length;

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee&hayes

limiting addressable memory to a range addressable by non-native program modules.

36. (**ORIGINAL**) A method as recited in claim 35, wherein the emulating step further comprises translating a non-native paradigm for passing parameters into a native paradigm for passing parameters.

37. (**ORIGINAL**) A method as recited in claim 34, wherein the converting step further comprises translating words from non-native word size into native word size.

38. (**ORIGINAL**) A computer comprising one or more computer-readable media having computer-executable instructions that, when executed by the computer, perform the method as recited in claim 34.

39. (**ORIGINAL**) A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 34.

40. (**ORIGINAL**) A kernel emulator configured to emulate a non-native kernel for a native computing platform so that kernel calls from non-native applications are translated into calls to a native kernel.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323-8979
www.leehayes.com
lee&hayes

41. (ORIGINAL) An emulator as recited in claim 40, wherein the emulator comprises:

an instruction-translator configured to translate non-native CPU instructions into native CPU instructions;

an address-translator configured to translate addresses from non-native length into native length;

an memory constrainer configured to limit addressable memory to a range addressable by non-native program modules.

42. (PREVIOUSLY PENDING) An operating system on a computer-readable medium, comprising:

a native kernel configured to receive calls from native program modules;

a kernel emulator as recited in claim 40 configured to receive calls from non-native program modules.

43. (CANCELED)

44. (CANCELED)

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee@hayes

45. (**ORIGINAL**) A kernel emulator for non-native program modules, the emulator comprising:

an interceptor configured to intercept kernel calls from non-native program modules;

a call-converter configured to convert non-native kernel calls intercepted by the interceptor into native kernel calls, wherein the call-converter comprises:

an instruction-translator configured to translate non-native CPU instructions into native CPU instructions;

an address-translator configured to translate addresses from non-native length into native length.

46. (**ORIGINAL**) An operating system on a computer-readable medium, comprising:

a native kernel configured to receive calls from native program modules;

a kernel emulator as recited in claim 45 configured to receive calls from non-native program modules.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com
lee&hayes